

Documento de propuesta de desarrollo
de software I, II y III

Dungeon **SCHOOL**

INTEGRANTES:

BRAYAN ANDRÉS RAMOS FUENTES



bramosfuentes42@correo.unicordoba.edu.co



EMILIO JOSÉ PLAZA VALDÉS



eplazavaldes@correo.unicordoba.edu.co



DIEGO ANDRÉS SALAS ALVARÉZ



dasalasalvarez@correo.unicordoba.edu.co



MARÍA LUCIA DILSO MOSQUERA



mdilsomosquera@correo.unicordoba.edu.co



TUTOR: ALEXANDER TOSCANO RICARDO

Descripción del videojuego educativo

Se busca crear y diseñar un videojuego original para ordenadores ya sea flash o instalable que permita la evaluación y diseño de estas en diferentes áreas centradas en temas específicos.

Dichas evaluaciones dependiendo el área estarán divididas en 2 sesiones donde en la primera se centrarán en preguntas esparcidas por un mapa que parodia a un salón de clases donde también abran ítems de ayuda de manera aleatoria y la segunda sección se centrará en la representación de un maestro como jefe final donde estará el grueso de la evaluación que se está aplicando de igual manera se podrán usar ítems sobrantes de primera sección en la parte del jefe.

El videojuego tendrá temas y preguntas prefabricadas y de diferentes dificultades de igual manera se buscará en próximas actualizaciones que estas puedan ser más personalizables al igual que los temas centrales o que se implemente la capacidad de integrada por IA de generar preguntas pertinentes en base a un plan de periodo o plan de clases.

ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS 5

1. INTRODUCCIÓN	5
PROPÓSITO DEL DOCUMENTO	5
ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA	7
DEFINICIONES Y ACRÓNIMOS.....	9
2. DESCRIPCIÓN GENERAL	9
OBJETIVOS DEL SISTEMA	9
FUNCIONALIDAD GENERAL.....	9
USUARIOS DEL SISTEMA	10
RESTRICCIONES	11
3. REQUISITOS FUNCIONALES	11
CASOS DE USO	11
DIAGRAMAS DE FLUJO DE CASOS DE USO	13
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO; ERROR! MARCADOR NO DEFINIDO.	
PRIORIDAD DE REQUERIMIENTOS.....	32
4. REQUISITOS NO FUNCIONALES	33
REQUISITOS DE DESEMPEÑO	33
REQUISITOS DE SEGURIDAD	33
REQUISITOS DE USABILIDAD.....	33
REQUISITOS DE ESCALABILIDAD	34
5. MODELADO E/R	35
DIAGRAMA DE ENTIDAD-RELACIÓN.....	35
DIAGRAMA RELACIONAL	36
SCRIPT DE MODELO RELACIONAL	37
DESCRIPCIÓN DE ENTIDADES Y RELACIONES.....	37
REGLAS DE INTEGRIDAD REFERENCIAL.....	39
COLECCIONES (NOSQL).....	40
6. ANEXOS.....	41
DIAGRAMAS ADICIONALES	41
REFERENCIAS	41
ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND	42
7. INTRODUCCIÓN	42
PROPÓSITO DE LA ETAPA	42
ALCANCE DE LA ETAPA	42
DEFINICIONES Y ACRÓNIMOS.....	42
8. DISEÑO DE LA ARQUITECTURA DE BACKEND	42
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA.....	42
COMPONENTES DEL BACKEND	42
DIAGRAMAS DE ARQUITECTURA	42
9. ELECCIÓN DE LA BASE DE DATOS	42
EVALUACIÓN DE OPCIONES (SQL o NoSQL)	42
JUSTIFICACIÓN DE LA ELECCIÓN	42
DISEÑO DE ESQUEMA DE BASE DE DATOS.....	42
10. IMPLEMENTACIÓN DEL BACKEND	42

ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN.....	42
CREACIÓN DE LA LÓGICA DE NEGOCIO	42
DESARROLLO DE ENDPOINTS Y APIs	43
AUTENTICACIÓN Y AUTORIZACIÓN	43
11. CONEXIÓN A LA BASE DE DATOS.....	43
CONFIGURACIÓN DE LA CONEXIÓN	43
DESARROLLO DE OPERACIONES CRUD	43
MANEJO DE TRANSACCIONES.....	43
12. PRUEBAS DEL BACKEND.....	43
DISEÑO DE CASOS DE PRUEBA	43
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN	43
MANEJO DE ERRORES Y EXCEPCIONES	43
ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND.....	44
13. INTRODUCCIÓN	44
PROPÓSITO DE LA ETAPA	44
ALCANCE DE LA ETAPA	44
DEFINICIONES Y ACRÓNIMOS.....	44
14. CREACIÓN DE LA INTERFAZ DE USUARIO (UI).....	44
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS	44
CONSIDERACIONES DE USABILIDAD	44
MAQUETACIÓN RESPONSIVA	44
15. PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS).....	44
DESARROLLO DE LA LÓGICA DEL FRONTEND	44
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	44
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APlicable).....	44
16. CONSUMO DE DATOS DESDE EL BACKEND	44
CONFIGURACIÓN DE CONEXIONES AL BACKEND	44
OBTENCIÓN Y PRESENTACIÓN DE DATOS	44
ACTUALIZACIÓN EN TIEMPO REAL (SI APlicable).....	45
17. INTERACCIÓN USUARIO-INTERFAZ.....	45
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS.....	45
IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS.....	45
MEJORAS EN LA EXPERIENCIA DEL USUARIO.....	45
18. PRUEBAS Y DEPURACIÓN DEL FRONTEND	45
DISEÑO DE CASOS DE PRUEBA DE FRONTEND.....	45
PRUEBAS DE USABILIDAD	45
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO	45
19. IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND	45
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) ...	45
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND	45
20. INTEGRACIÓN CON EL BACKEND.....	45
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND	45
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND.....	45
ANEXOS.....	46

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

1. Introducción

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación del videojuego educativo en su etapa inicial. El desarrollo se basará en la metodología SECMALI, la cual permitirá estructurar y validar cada fase del proyecto para garantizar su efectividad pedagógica y técnica.

El documento se divide en tres etapas para facilitar su entendimiento y aplicación en la asignatura de diseño de software.

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Diseño de la Aplicación y Análisis de Requisitos. Esta fase se enfoca en definir los elementos fundamentales del videojuego educativo, considerando su estructura, mecánicas y requerimientos técnicos. El objetivo es establecer una base sólida que permita desarrollar una experiencia lúdica centrada en el aprendizaje y la evaluación de conocimientos en un entorno simulado tipo aula escolar.

El videojuego estará dividido en dos secciones principales:

1. Exploración del mapa:

El jugador recorrerá un entorno en pixel art que representa un aula de clases. A lo largo del mapa encontrará preguntas educativas de dificultad variada, así como ítems de ayuda dispersos aleatoriamente que podrán ser usados más adelante.

2. Enfrentamiento con el maestro (jefe final):

Tras completar la exploración, el jugador se enfrentará al “jefe final”, representado por un maestro. En esta fase se evaluarán los conocimientos adquiridos mediante preguntas más complejas, y se podrán utilizar los ítems recolectados previamente para facilitar el progreso.

Objetivos del diseño:

- Evaluar el aprendizaje de los estudiantes de forma dinámica y entretenida.
- Fomentar el uso de la tecnología como recurso pedagógico.
- Motivar a los usuarios mediante retos progresivos y retroalimentación visual.
- Crear un entorno inmersivo adaptado a la estética pixel art.

Requerimientos funcionales:

- Sistema de preguntas por niveles o temas.
- Registro de respuestas correctas e incorrectas.
- Uso estratégico de ítems recolectados.
- Interfaz intuitiva para navegación y combate.
- Escenarios interactivos con diseño en pixel art.

Requerimientos no funcionales:

- Compatibilidad con ordenadores de bajos recursos.
- Bajo consumo de ancho de banda en caso de conexión al servidor.
- Facilidad de actualización y personalización de preguntas.
- Diseño atractivo y accesible para estudiantes de distintos niveles.

Etapa 2: Persistencia de Datos con Backend – Servidor

En esta etapa se desarrollará el backend que permitirá almacenar y gestionar la información generada durante el uso del videojuego. Esto incluye el progreso de los jugadores, respuestas seleccionadas, estadísticas de juego y configuraciones personalizadas.

Objetivos:

- Crear una API que permita el almacenamiento y recuperación de datos del juego.
- Implementar una base de datos para guardar el progreso del jugador, resultados por nivel, preguntas y configuración de ítems.
- Facilitar la personalización futura del contenido por parte de docentes (subida de preguntas propias, configuraciones de dificultad).

Componentes:

- Endpoints para el registro y login de usuarios.
- Registro de progreso por sesión.
- Almacenamiento de preguntas y resultados por jugador.
- Carga dinámica de datos (preguntas, niveles, ítems).

Tecnologías sugeridas:

- Lenguaje: JavaScript o TypeScript.
- Framework: Express.js o Nest.js.
- Base de datos: MongoDB (NoSQL) o PostgreSQL (SQL), según necesidades.
- API: RESTful con autenticación mediante JWT.
- Herramientas adicionales: Mongoose (si se usa MongoDB), Prisma (si se usa PostgreSQL).

Etapa 3: Consumo de Datos y Desarrollo Frontend – Cliente

Esta etapa contempla la implementación de la interfaz visual y funcional del videojuego en estilo pixel art, centrado en brindar una experiencia lúdica y educativa atractiva.

Objetivos:

- Desarrollar una interfaz gráfica en pixel art que represente un aula y escenarios de batalla con el jefe final.
- Implementar mecánicas de exploración, interacción con preguntas y uso de ítems.
- Conectar el cliente con el backend para cargar preguntas, enviar respuestas y registrar resultados.

Características:

- Sección de exploración estilo RPG 2D con estética pixel art.
- Mecánicas para recolectar ítems y resolver preguntas en un entorno interactivo.
- Sección de jefe final donde se aplican preguntas más complejas.
- Uso de los ítems recolectados como ayudas o potenciadores.
- Feedback visual retro acorde a la estética pixel art (barras de energía, cuadros de texto tipo RPG).

Tecnologías sugeridas:

- Motor/Framework para 2D en pixel art: Phaser.js o Godot (GDScript) si se quiere mantener libre de Unity.
- Lenguaje: JavaScript (Phaser) o GDScript (Godot).
- Recursos visuales: Sprites 2D en pixel art, tilesets para el mapa del aula, animaciones sencillas.
- Sonido: Efectos retro y música chiptune para ambientar.

Alcance del Proyecto Dungeon School

Dungeon School es una propuesta innovadora de videojuego educativo multiplataforma que busca revolucionar el aprendizaje en entornos digitales. Mezclando mecánicas de juegos de rol con elementos pedagógicos, este proyecto se plantea como una herramienta divertida e interactiva para reforzar conocimientos en el aula o desde casa. Ambientado en un universo escolar con estética pixel art retro, el juego permitirá a los estudiantes recorrer mapas tipo aula, resolver retos académicos, recolectar ítems útiles y enfrentarse a un "maestro jefe" para demostrar lo aprendido.

El objetivo principal es motivar a los estudiantes a aprender jugando, brindando una experiencia inmersiva y significativa. Además, el juego está diseñado para expandirse con el tiempo, integrando funciones cada vez más complejas que permitan adaptar el contenido a distintos niveles educativos, personalizar la experiencia del usuario y fomentar el trabajo colaborativo. Este documento detalla el alcance inicial y propone una clasificación de funcionalidades futuras según la complejidad que implicaría su desarrollo e implementación.

Funcionalidades actuales:

- Movimiento del personaje (Arriba, abajo, izquierda, derecha)
- Desbloqueo de zonas
- Interacción con preguntas
- Recolección de ítems
- Uso de ítems
- Enfrentamiento con un enemigo
- Registro de respuestas
- Estado del personaje (barras de energía, barra de estado, etc.)

Funcionalidades futuras:

Funciones Básicas (fácil implementación)

- Guardar el progreso del jugador automáticamente.
- Personalizar el avatar del estudiante con ropa o accesorios.
- Elegir entre distintos mapas escolares para explorar.
- Obtener pistas al ver contenido educativo dentro del juego.
- Subir puntuaciones a un ranking global o escolar.
- Conseguir medallas por logros educativos.
- Cambiar la dificultad del juego según el nivel del estudiante.
- Escuchar narraciones o audios explicativos como ayuda extra.
- Compartir resultados o certificados con padres o docentes.
- Añadir minijuegos entre preguntas para mantener la motivación.
- Personalizar el fondo o estilo del aula virtual.
- Acceder a estadísticas personales de rendimiento.
- Guardar partidas y continuar desde donde se dejó.

Funciones Medias (requieren planificación y desarrollo intermedio)

- Sistema de progresión con logros.
- Versión portable offline.
- Desbloquear niveles temáticos (ciencias, historia, etc.).
- Usar objetos especiales para resolver preguntas más rápido.
- Participar en torneos académicos entre colegios.

- Chatear con otros jugadores dentro del aula virtual.
- Usar animaciones o reacciones del maestro jefe según el rendimiento.
- Integrar videos educativos antes de ciertas preguntas clave.
- Permitir a los profes crear mini misiones con recompensas.
- Recibir retroalimentación automática según errores comunes.
- Jugar con teclado, mouse, pantalla táctil o control.
- Descargar preguntas desde una nube educativa segura.
- Acceder a una biblioteca con guías de estudio integradas.
- Usar un sistema de energía para regular el tiempo de juego.
- Hacer misiones diarias o semanales con objetivos específicos.

Funciones Complejas (altamente personalizadas o tecnológicamente avanzadas)

- Personalización de pruebas por parte del docente.
- Generación automática de preguntas con IA según el plan de clase.
- Modo multijugador (en tiempo real o por turnos).
- Participar en eventos mensuales con recompensas especiales.
- Usar comandos por voz para responder o navegar.
- Desbloquear zonas secretas si se alcanza cierto puntaje.
- Usar monedas virtuales para desbloquear contenido educativo extra.

Definiciones y Acrónimos

- API: Interfaz de Programación de Aplicaciones (Application Programming Interface).
- DBMS: Sistema de Gestión de Bases de Datos (Database Management System).
- JWT: Token Web JSON (JSON Web Token), utilizado para autenticar usuarios.
- CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).
- Pixel Art: Estilo gráfico basado en píxeles visibles, común en videojuegos retro.
- RPG: Juego de Rol (Role-Playing Game), estilo de juego donde el jugador asume el papel de un personaje en una narrativa.

2. Descripción General

Objetivos del Sistema

El objetivo del sistema es proporcionar una pizarra compartida dentro de un Sistema de Gestión de Contenido llamado CREA VI que permita a los usuarios colaborar de manera eficiente y efectiva, facilitando la creación, visualización y edición de contenido visual en tiempo real. Esta pizarra compartida se diseñará con el propósito de mejorar la comunicación y la colaboración en un entorno en línea, ofreciendo a los usuarios una plataforma intuitiva y versátil para crear y compartir ideas, diagramas, esquemas y contenido visual de manera colaborativa, enriqueciendo así la experiencia de usuario y la productividad en el uso del CMS.

Funcionalidad General

- **Creación y Edición Colaborativa:** Permite a los usuarios crear y editar contenido en la pizarra de forma colaborativa en tiempo real. Múltiples usuarios pueden trabajar en el mismo documento simultáneamente.
- **Herramientas de Dibujo y Anotación:** Proporciona herramientas de dibujo, pinceles, formas y opciones de anotación que permiten a los usuarios plasmar sus ideas y conceptos de manera visual.
- **Carga de Imágenes y Multimedia:** Permite a los usuarios cargar imágenes, videos y otros medios directamente en la pizarra, lo que facilita la ilustración de conceptos.
- **Organización de Contenido:** Ofrece opciones para organizar y estructurar el contenido en la pizarra, como la creación de capas, agrupación de elementos y uso de etiquetas.
- **Historial de Revisiones:** Registra un historial de revisiones que permite a los usuarios rastrear los cambios realizados en la pizarra y restaurar versiones anteriores si es necesario.
- **Compartir y Colaborar:** Permite compartir la pizarra con otros usuarios a través de enlaces o invitaciones, lo que facilita la colaboración con colegas, clientes o amigos.
- **Comentarios y Chat en Tiempo Real:** Los usuarios pueden comentar y discutir sobre el contenido de la pizarra a través de un chat en tiempo real, lo que facilita la comunicación durante la colaboración.
- **Exportación e Impresión:** Ofrece la capacidad de exportar el contenido de la pizarra en varios formatos (PDF, imagen, etc.) y la opción de imprimirla.
- **Integración con el CMS:** Se integra de manera transparente con el sistema de gestión de contenido (CMS CREA VI), lo que permite incrustar pizarras en los contenidos, metodologías o cualquier otro tipo de componente que permita la pizarra.
- **Personalización y Temas:** Permite a los usuarios personalizar la apariencia de la pizarra y seleccionar temas que se adapten a sus necesidades.
- **Acceso Seguro:** Proporciona medidas de seguridad para garantizar que solo los usuarios autorizados puedan acceder y editar la pizarra.
- **Notificaciones y Actualizaciones en Tiempo Real:** Los usuarios reciben notificaciones sobre cambios en la pizarra y pueden ver actualizaciones en tiempo real mientras otros editan.
- **Acceso Móvil:** Ofrece una experiencia de usuario optimizada en dispositivos móviles, permitiendo el acceso y la colaboración desde smartphones y tabletas.
- **Búsqueda y Filtros:** Facilita la búsqueda de contenido en la pizarra y la aplicación de filtros para organizar y encontrar información específica.
- **Gestión de Usuarios y Permisos:** Permite a los administradores gestionar usuarios y definir permisos de acceso y edición.
- **Informes y Analíticas:** Proporciona información sobre el uso de la pizarra, como quién la ha editado, cuándo se realizaron cambios y estadísticas sobre el contenido(XAPI).

Usuarios del Sistema

Los siguientes usuarios pueden interactuar con la pizarra dependiendo de las funcionalidades.

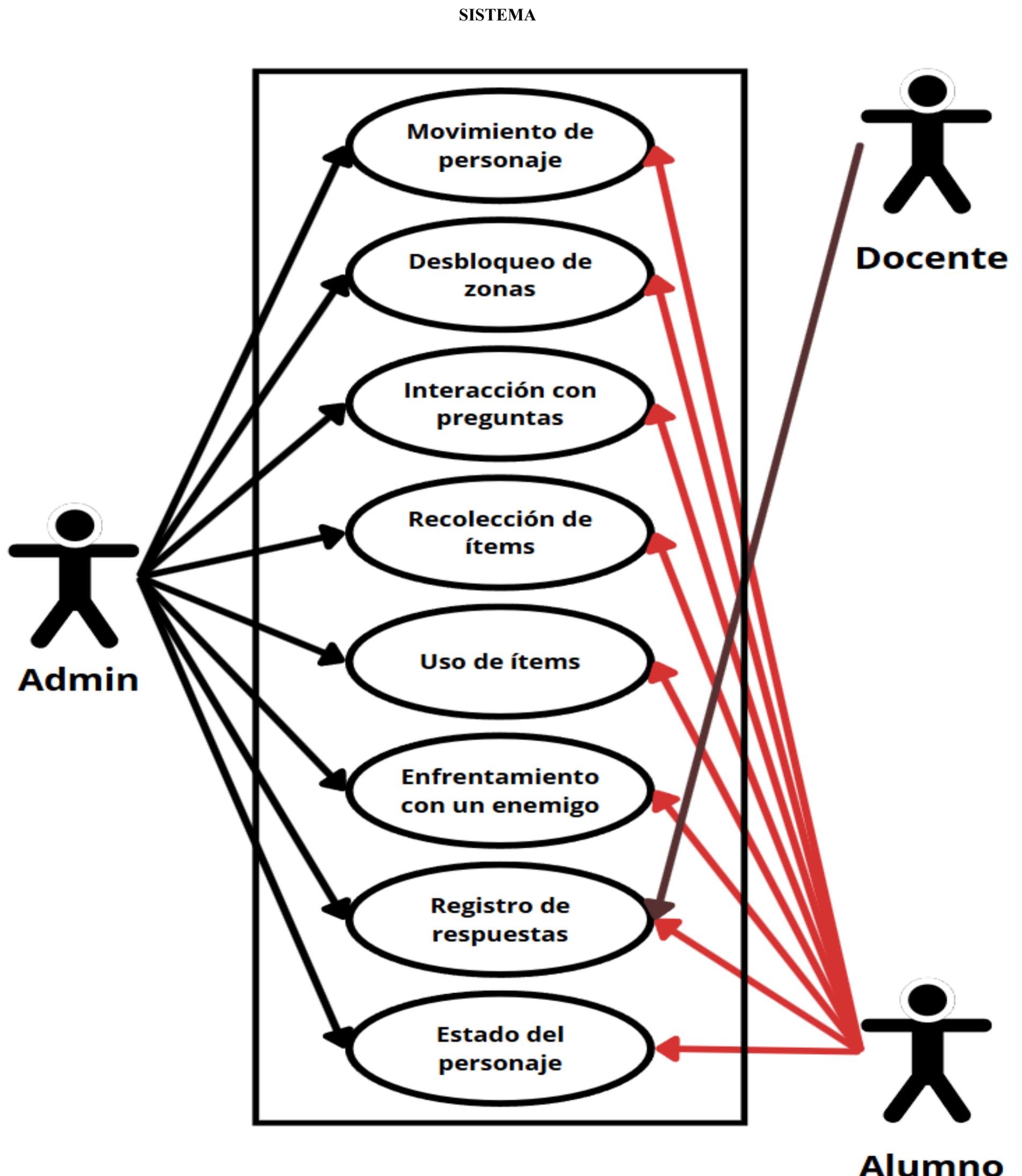
Funcionalidad	Administradores	Docente	Alumno
Movimiento del personaje (Arriba, abajo, izquierda, derecha)	✓		✓
Desbloqueo de zonas	✓		✓
Interacción con preguntas	✓		✓
Recolección de ítems	✓		✓
Uso de ítems	✓		✓
Enfrentamiento con un enemigo	✓		✓
Registro de respuestas	✓	✓	✓
Estado del personaje (barras de energía, barra de estado, etc.)	✓		✓
Guardar el progreso	✓		✓
Personalizar el avatar	✓		✓
Unirse a una partida	✓	✓	✓
Crear nueva partida	✓	✓	
Elegir entre distintos mapas	✓	✓	✓
Obtener pistas	✓		✓
Ranking	✓	✓	✓
Conseguir medallas	✓		✓
Cambiar la dificultad	✓	✓	
Escuchar narraciones	✓		✓
Compartir resultados	✓	✓	✓
Continuar partida	✓	✓	✓
Personalizar el fondo	✓	✓	
Acceder a estadísticas	✓	✓	
Guardar partidas	✓	✓	✓
Crear sesión	✓	✓	✓
Cerrar sesión	✓	✓	✓

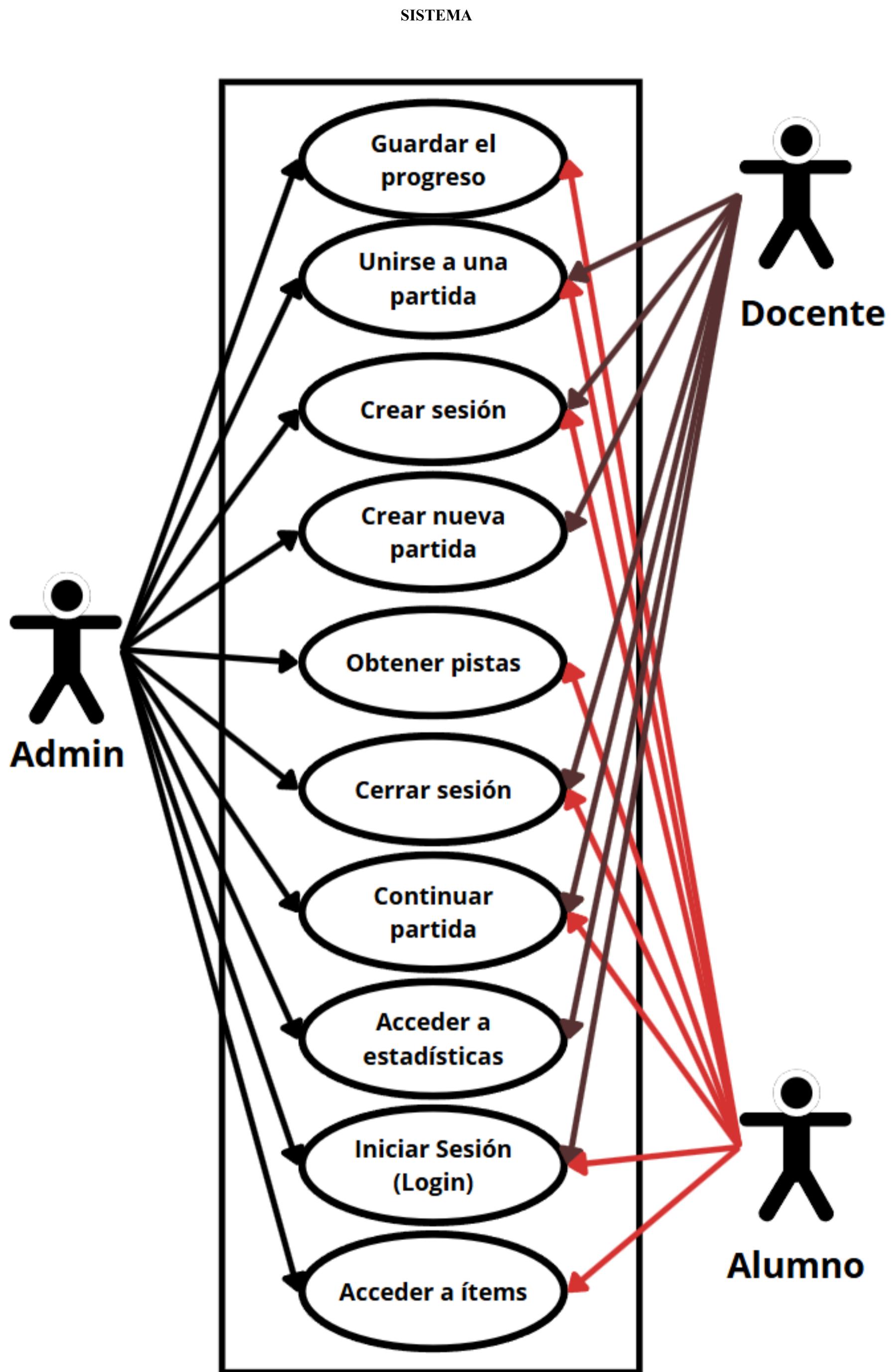
Restricciones

3. Requisitos Funcionales

Casos de Uso

Diagrama de caso de uso





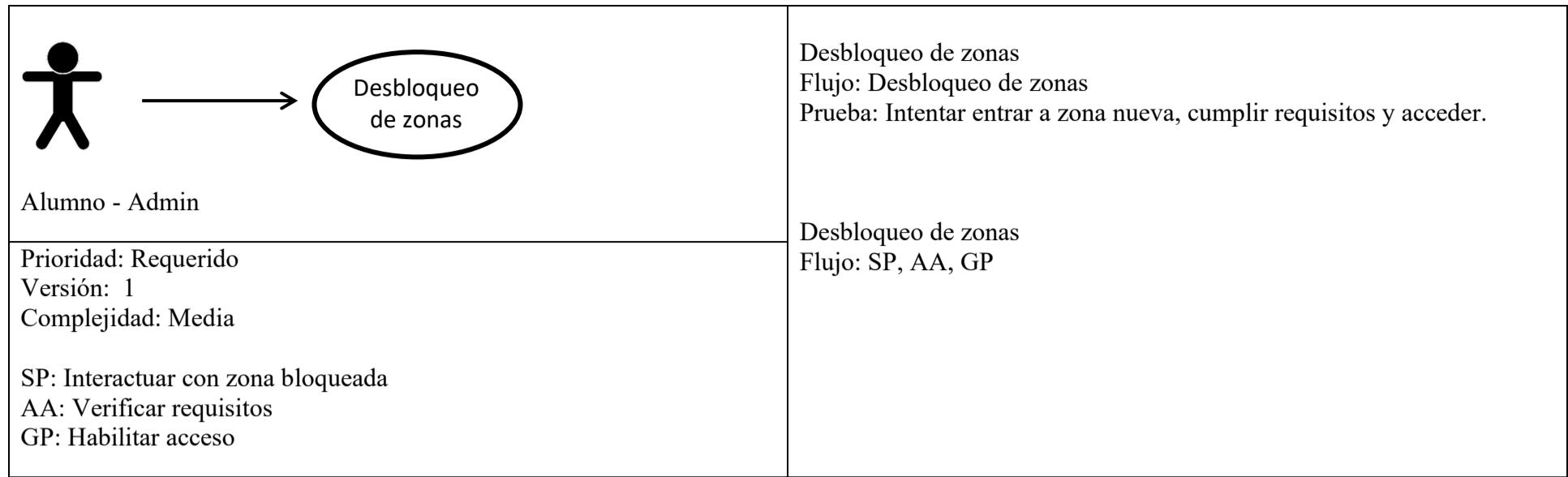
Diagramas de Flujo de Casos de Uso y descripción detallada de cada caso de uso

CASO N°1 Movimiento del personaje

	Movimiento del personaje	<p>1. Movimiento del personaje Flujo: Movimiento del personaje Prueba: Desplazarse por el mapa, interactuar con obstáculos y zonas nuevas.</p> <p>Movimiento del personaje Flujo: SP, AA, GP</p>
Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Presionar tecla de dirección AA: Actualizar posición GP: Verificar colisiones		

ID	CU-ALU-001	
Nombre	Movimiento del personaje	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno moverse por el entorno del aula virtual mediante las teclas de dirección.	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	El alumno ha iniciado sesión y se encuentra en el entorno de juego.	
Flujo Normal	Alumno	Sistema
	Presiona una tecla de dirección	
		Detecta la dirección solicitada
		Valida la nueva posición del personaje
		Actualiza la posición del personaje
		Retorna la posición del personaje
		Refleja visualmente el cambio en pantalla
Flujo alternativo 1	Presiona varias teclas a la vez	
		El sistema prioriza una dirección
		Muestra la nueva posición
Flujo alternativo 2	Tecla inválida o sin función	
		Sistema no realiza ninguna acción
Post-condiciones	El personaje se mueve correctamente por el escenario sin errores. El personaje nunca sobrepasa la zona delimitada.	
Excepciones	Error en la carga del escenario o glitch en colisiones.	

CASO N°2 Desbloqueo de zonas



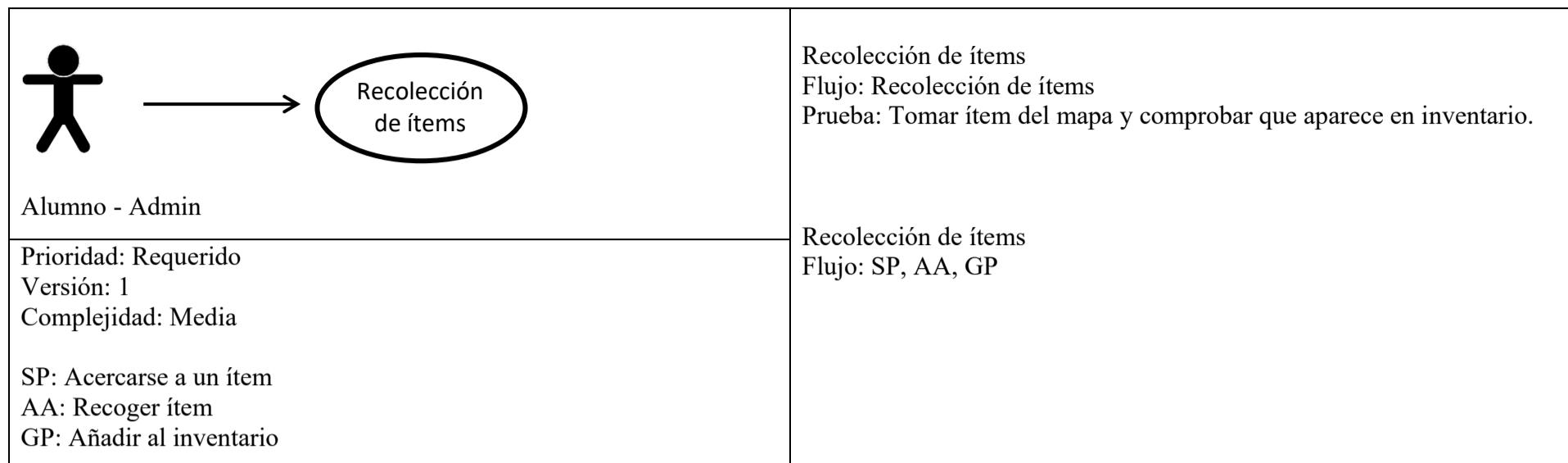
ID	CU-ALU-002	
Nombre	Desbloqueo de zonas	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno acceder a nuevas zonas del mapa al cumplir ciertos requisitos.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El alumno ha recolectado los ítems o respondido las preguntas necesarias.	
Flujo Normal	Alumno	Sistema
	Se ubica en una zona bloqueada	
		Verifica requisitos cumplidos
		Habilita la entrada a la nueva zona
		Reproduce animación de desbloqueo
		Se desbloquea la zona
Flujo alternativo 1	Intenta entrar sin cumplir requisitos	
		Muestra mensaje de acceso denegado
Flujo alternativo 2	Zona desbloqueada previamente	
		Acceso inmediato sin validación
Post-condiciones	El alumno accede correctamente a una nueva sección del juego.	
Excepciones	Bug de colisión impide entrada o error en validación de requisitos.	

CASO N°3 Interacción con preguntas

 → Interacción con preguntas	<p>Interacción con preguntas Flujo: Interacción con preguntas Prueba: El alumno se acerca a la pregunta, selecciona una opción.</p> <p>Interacción con preguntas Flujo: SP, AA, GP</p>
<p>Alumno - Admin</p> <p>Prioridad: Requerido Versión: 1 Complejidad: Media</p> <p>SP: Seleccionar pregunta AA: Elegir respuesta GP: Evaluar y registrar respuesta</p>	

ID	CU-ALU-003	
Nombre	Interacción con preguntas	
Actores	Alumno, Sistema	
Objetivo	Interactuar con preguntas permite al alumno seleccionar y responder las preguntas durante la exploración del juego.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El alumno se encuentra frente a la pregunta visible en el mapa.	
Flujo Normal	Alumno	Sistema
	Selecciona una pregunta	
		Muestra la interfaz de respuestas
	Elige una respuesta	
		Bloquea seleccionador de respuesta
		Evalúa si es correcta o incorrecta
		Registra la respuesta en la base de datos
Flujo alternativo 1	No selecciona ninguna opción	
		Bloquea el botón de "Enviar"
Flujo alternativo 2	Responde incorrectamente	
		Registra la respuesta en la base de datos
Post-condiciones	La respuesta queda registrada y el sistema guarda el resultado.	
Excepciones	Error de conexión con la base de datos o pregunta mal cargada.	

CASO N°4 Recolección de ítems



ID	CU-ALU-004	
Nombre	Recolección de ítems	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno recoger ítems del entorno para su posterior uso.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El ítem está visible en el entorno y el jugador puede recolectarlo.	
Flujo Normal	Alumno	Sistema
	Se acerca al ítem	
		Detecta la proximidad
	Presiona la tecla de acción	
		El ítem desaparece del entorno
		Añade el ítem al inventario
		Muestra notificación de recolección
Flujo alternativo 1	El inventario está lleno	
		Muestra mensaje de espacio insuficiente
Flujo alternativo 2	El ítem ya fue recogido antes	
		No ocurre ninguna acción
Post-condiciones	El ítem queda guardado en el inventario del jugador.	
Excepciones	Error en la animación de recogida o ítem duplicado por bug.	

CASO N°5 Uso de ítems

 → Uso de ítems	Uso de ítems Flujo: Uso de ítems Prueba: Usar un ítem desde el inventario y ver su efecto reflejado.
Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Abrir inventario AA: Seleccionar ítem GP: Aplicar efecto y actualizar	Uso de ítems Flujo: SP, AA, GP

ID	CU-ALU-005	
Nombre	Uso de ítems	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno utilizar ítems recolectados para obtener ventajas en el juego.	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	El alumno debe tener al menos un ítem en el inventario.	
Flujo Normal	Alumno	Sistema
	Abre el inventario	
		Muestra lista de ítems disponibles
	Selecciona un ítem	
		Se verifica el tipo de ítem
	Confirma el uso del ítem	
		Aplica efecto correspondiente
		Actualiza estado del personaje
		Muestra mensaje de confirmación
Flujo alternativo 1	El ítem no es aplicable en ese momento	
		Muestra advertencia "No se puede usar aquí"
Flujo alternativo 2	El ítem ya fue usado o está agotado	
		Muestra mensaje "Ítem no disponible"
Post-condiciones	El efecto del ítem se aplica correctamente y se actualiza el inventario.	
Excepciones	Error en la base de datos o bug que impide el uso correcto del ítem.	

CASO N° 6 Enfrentamiento con el jefe final

 → Enfrentamiento con el jefe final	Enfrentamiento con un enemigo Flujo: Enfrentamiento con un enemigo Prueba: Completar preguntas del jefe final usando ítems recolectados.
Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Alta SP: Iniciar enfrentamiento AA: Contestar preguntas GP: Evaluar desempeño y actualizar estado	Enfrentamiento con un enemigo Flujo: SP, AA, GP

ID	CU-ALU-006	
Nombre	Enfrentamiento con el jefe final	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno enfrentarse al jefe final respondiendo preguntas y utilizando ítems.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	El alumno ha completado la fase de exploración y ha recolectado ítems.	
Flujo Normal	Alumno	Sistema
	Inicia enfrentamiento	
		Carga animación del enemigo
	Responde preguntas complejas	
		Evalúa respuestas en tiempo real
	Usa ítems como ayuda	
		Aplica efectos si son válidos
		Muestra retroalimentación visual
		Determina resultado del enfrentamiento
Flujo alternativo 1	Responde incorrectamente varias preguntas	
		Muestra barra de energía reducida
Flujo alternativo 2	Agota sus ítems y continúa	
		Se evalúa solo con conocimientos
Post-condiciones	Se actualiza el estado del juego según el resultado (victoria o derrota).	
Excepciones	Errores de carga de preguntas o ítems no funcionales durante el combate.	

CASO N°7 Registro de respuestas

 → Registro de respuestas	Registro de respuestas Flujo: Registro de respuestas Prueba: Responder y verificar que la base de datos almacene el resultado.
Alumno - Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Responder pregunta AA: Registrar respuesta GP: Asociar con usuario	Registro de respuestas Flujo: SP, AA, GP

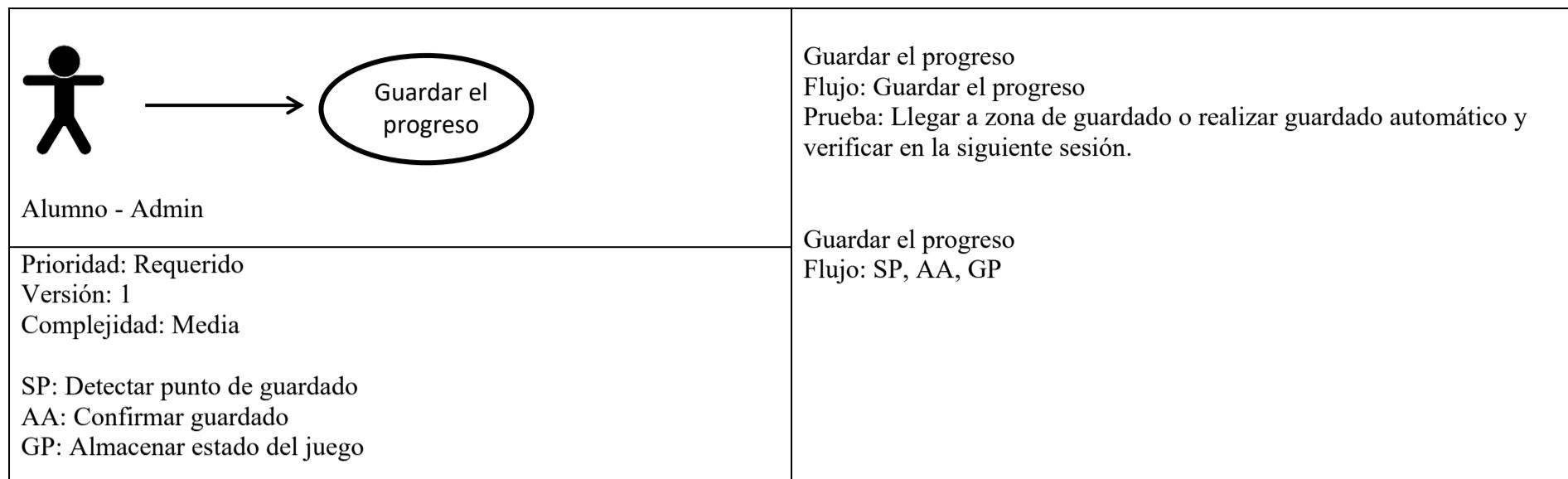
ID	CU-ALU-007	
Nombre	Registro de respuestas	
Actores	Alumno, Sistema	
Objetivo	Permitir al sistema almacenar las respuestas seleccionadas por los alumnos para su posterior análisis.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El alumno ha respondido al menos una pregunta durante el juego.	
Flujo Normal	Alumno	Sistema
	Envía respuesta seleccionada	
		Recibe respuesta seleccionada del alumno
		Registra respuesta en la base de datos
		Retorna mensaje de respuesta guardada
Flujo alternativo 1	El alumno cierra el juego antes de responder	
		La respuesta no se registra
Flujo alternativo 2	La conexión se pierde durante el envío	
		Se reintenta guardar automáticamente
Post-condiciones	La respuesta queda almacenada correctamente y disponible para ser calificadas.	
Excepciones		

CASO N°8 Estado del personaje

 → Estado del personaje	Estado del personaje Flujo: Estado del personaje Prueba: Recibir daño o beneficio y observar cambios en barras visuales.
Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Baja SP: Cambios en juego AA: Actualizar barra de estado GP: Mostrar en pantalla	Estado del personaje Flujo: SP, AA, GP

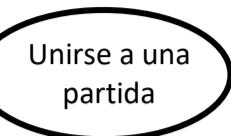
ID	CU-ALU-008	
Nombre	Estado del personaje	
Actores	Alumno, Sistema	
Objetivo	Visualizar en pantalla el estado actual del personaje (energía, barra de progreso, etc.).	
Urgencia	3	
Esfuerzo	1	
Pre-condiciones	El alumno debe estar en una partida activa.	
Flujo Normal	Alumno	Sistema
	Interactúa con el entorno	
		Detecta cambios en el estado (daño, progreso)
		Almacenar en el sistema
		Retornar actualización del estado
		Actualiza visualmente las barras y atributos
		Refleja cambios en tiempo real en pantalla
Flujo alternativo 1	Completa una tarea y sube de nivel	
		Aumenta energía y actualiza barra de estado
Flujo alternativo 2	Nota que la barra no refleja los cambios recientes	
		Detecta error de sincronización
		Recupera el estado correcto
		Actualiza visualmente la barra de estado
Post-condiciones	El estado del personaje queda sincronizado con su estado actual.	
Excepciones	No se actualizan las barras por error en sprites o animaciones.	

CASO N°9 Guardar el progreso



ID	CU-ALU-009	
Nombre	Guardar el progreso	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno guardar su progreso actual en el juego.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El alumno debe estar en una sesión activa del juego.	
Flujo Normal	Alumno	Sistema
	Llega a un punto de guardado o abre el menú	
		Muestra opción de guardar
	Selecciona “Guardar progreso”	
		Detecta y recopila datos actuales
	Confirma guardado	
		Almacena datos en la base de datos
		Muestra mensaje de confirmación
Flujo alternativo 1	Cierra el juego sin guardar manualmente	
		Se activa guardado automático
Flujo alternativo 2	Conexión débil al guardar	
		Se reintenta guardar en segundo plano
Post-condiciones	El progreso del alumno queda guardado correctamente para futuras sesiones.	
Excepciones	Fallo en la conexión o error en la base de datos durante el guardado.	

CASO N° 10 Unirse a una partida

 → 	Unirse a una partida Flujo: Unirse a una partida Prueba: Elegir una partida activa y validar ingreso exitoso.
Alumno - Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Ver partidas disponibles AA: Seleccionar partida GP: Unirse a la sesión	Unirse a una partida Flujo: SP, AA, GP

ID	CU-ALDOC-001	
Nombre	Unirse a una partida	
Actores	Alumno, Docente, Sistema	
Objetivo	Permitir a los usuarios unirse a una partida activa previamente creada.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	Debe haber una partida disponible para unirse.	
Flujo Normal	Alumno/Docente	Sistema
	Accede al menú de partidas	
		Muestra lista de partidas disponibles
	Selecciona una partida	
		Verifica disponibilidad
	Confirma unirse	
		Asocia usuario a la partida
		Carga escenario de la partida
Flujo alternativo 1	La partida está llena	
		Muestra mensaje "Partida no disponible"
Flujo alternativo 2	Partida no encontrada o eliminada	
		Muestra advertencia y actualiza lista
Post-condiciones	El usuario se une correctamente a la partida seleccionada.	
Excepciones	Error de sincronización o fallo en carga del entorno.	

CASO N° 11 Crear sesión

 → Crear sesión	Crear sesión Flujo: Crear sesión Prueba: Ingresar usuario/contraseña válidos y acceder al juego.
Alumno - Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Ingresar datos AA: Validar información GP: Generar sesión activa	Crear sesión Flujo: SP, AA, GP

ID	CU-ALDOC-002	
Nombre	Crear sesión	
Actores	Alumno, Docente, Sistema	
Objetivo	Permitir a los usuarios iniciar sesión en el sistema para acceder a sus partidas y funciones.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El usuario debe tener una cuenta registrada.	
Flujo Normal	Alumno/Docente	Sistema
	Abre el menú de inicio de sesión	
		Muestra formulario de login
	Ingresar usuario y contraseña	
		Verifica credenciales
		Crea sesión activa
		Redirige al menú principal
Flujo alternativo 1	Olvida la contraseña	
		Opción de recuperación disponible
Flujo alternativo 2	Credenciales incorrectas	
		Muestra mensaje de error y solicita nuevo intento
Post-condiciones	El usuario inicia sesión y accede a su perfil y progreso.	
Excepciones	Error en servidor de autenticación o token inválido.	

CASO N°12 Crear nueva partida

 → Crear nueva partida	Crear nueva partida Flujo: Crear nueva partida Prueba: Configurar una nueva partida y verificar que inicie correctamente.
Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Definir parámetros de juego AA: Confirmar creación GP: Iniciar nueva partida	Crear nueva partida Flujo: SP, AA, GP

ID	CU-DOC-002	
Nombre	Crear nueva partida	
Actores	Docente, Sistema	
Objetivo	Permitir al docente crear una nueva partida para que los estudiantes puedan unirse.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El docente ha iniciado sesión correctamente.	
Flujo Normal	Docente	Sistema
	Accede al menú de creación de partida	
		Muestra formulario de configuración
	Configura parámetros (nombre, dificultad, etc.)	
		Verifica validez de datos
	Confirma creación	
		Registra la partida en la base de datos
		Muestra mensaje de partida creada con éxito
Flujo alternativo 1	Intenta crear una partida con nombre duplicado	
		Muestra advertencia y solicita cambio
Flujo alternativo 2	No completa los campos requeridos	
		Muestra mensaje de error y bloquea creación
Post-condiciones	La nueva partida queda disponible para ser unida por los estudiantes.	
Excepciones	Fallo en el servidor o error al registrar datos de la partida.	

CASO N°13 Obtener pistas

 → Obtener pistas	Obtener pistas Flujo: Obtener pistas Prueba: Solicitar ayuda en una pregunta y visualizar la pista correspondiente.
Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Baja SP: Activar contenido educativo AA: Esperar reproducción o interacción GP: Mostrar pista al jugador	Obtener pistas Flujo: SP, AA, GP

ID	CU-ALU-010	
Nombre	Obtener pistas	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno visualizar pistas para responder preguntas o avanzar en el juego.	
Urgencia	5	
Esfuerzo	1	
Pre-condiciones	El alumno debe estar interactuando con una pregunta o desafío.	
Flujo Normal	Alumno	Sistema
	Solicita ayuda o pista	
		Verifica si hay pista disponible
		Muestra contenido de pista (texto, audio o imagen)
		Permite volver a la pregunta
Flujo alternativo 1	No hay pista disponible para esa pregunta	
		Muestra mensaje: "No disponible"
Flujo alternativo 2	El alumno ya usó la pista	
		Muestra la pista nuevamente como referencia
Post-condiciones	El alumno recibe una pista válida que lo guía en su respuesta.	
Excepciones	Fallo en la carga del contenido multimedia o conexión interrumpida.	

CASO N°14 Cerrar sesión

 → Cerrar sesión	Cerrar sesión Flujo: Cerrar sesión Prueba: Presionar "Cerrar sesión" y verificar redirección al inicio.
Alumno - Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Baja SP: Seleccionar opción de cerrar sesión AA: Confirmar salida GP: Finalizar sesión y volver al menú	Cerrar sesión Flujo: SP, AA, GP

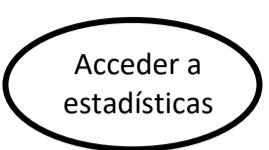
ID	CU-ALDOC-003	
Nombre	Cerrar sesión	
Actores	Alumno, Docente, Sistema	
Objetivo	Permitir al usuario salir del sistema de forma segura.	
Urgencia	5	
Esfuerzo	1	
Pre-condiciones	El usuario debe tener una sesión activa.	
Flujo Normal	Alumno/Docente	Sistema
	Accede al menú de usuario	
		Muestra opciones de cuenta
	Selecciona "Cerrar sesión"	
		Cierra sesión activa
		Redirige al menú de inicio de sesión
Flujo alternativo 1	Intenta cerrar sesión sin conexión	
		Cierra sesión localmente y sincroniza al reconnectar
Flujo alternativo 2	No confirma la acción	
		Cancela el proceso y permanece en sesión
Post-condiciones	La sesión se cierra y los datos quedan protegidos.	
Excepciones	Fallo en el cierre de sesión por error de servidor o token inválido.	

CASO N°15 Continuar partida

 → Continuar partida	<p>Continuar partida Flujo: Continuar partida Prueba: Iniciar desde último punto guardado y validar datos cargados.</p> <p>Continuar partida Flujo: SP, AA, GP</p> <p>Alumno - Docente - Admin</p> <p>Prioridad: Requerido Versión: 1 Complejidad: Media</p> <p>SP: Acceder a menú de partidas guardadas AA: Seleccionar partida GP: Cargar progreso</p>
--	--

ID	CU-ALDOC-004	
Nombre	Continuar partida	
Actores	Alumno, Docente, Sistema	
Objetivo	Permitir a los usuarios reanudar una partida previamente guardada.	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	Debe existir una partida guardada asociada al usuario.	
Flujo Normal	Alumno/Docente	Sistema
	Accede al menú de partidas guardadas	
		Muestra lista de partidas anteriores
	Selecciona una partida	
		Recupera datos de guardado
	Confirma reanudar	
Flujo alternativo 1	No hay partidas guardadas	
		Muestra mensaje: "No hay partidas disponibles"
Flujo alternativo 2	La partida seleccionada está corrupta	
		Muestra error y ofrece volver al menú
Post-condiciones	El usuario continúa desde el último punto guardado.	
Excepciones	Error de sincronización o fallo en la carga del estado del juego.	

CASO N°16 Acceder a estadísticas

 → 	Acceder a estadísticas Flujo: Acceder a estadísticas Prueba: Iniciar sesión como docente y ver gráficas de rendimiento. Acceder a estadísticas Flujo: SP, AA, GP
Docente - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Iniciar sesión AA: Navegar al panel de estadísticas GP: Visualizar datos	

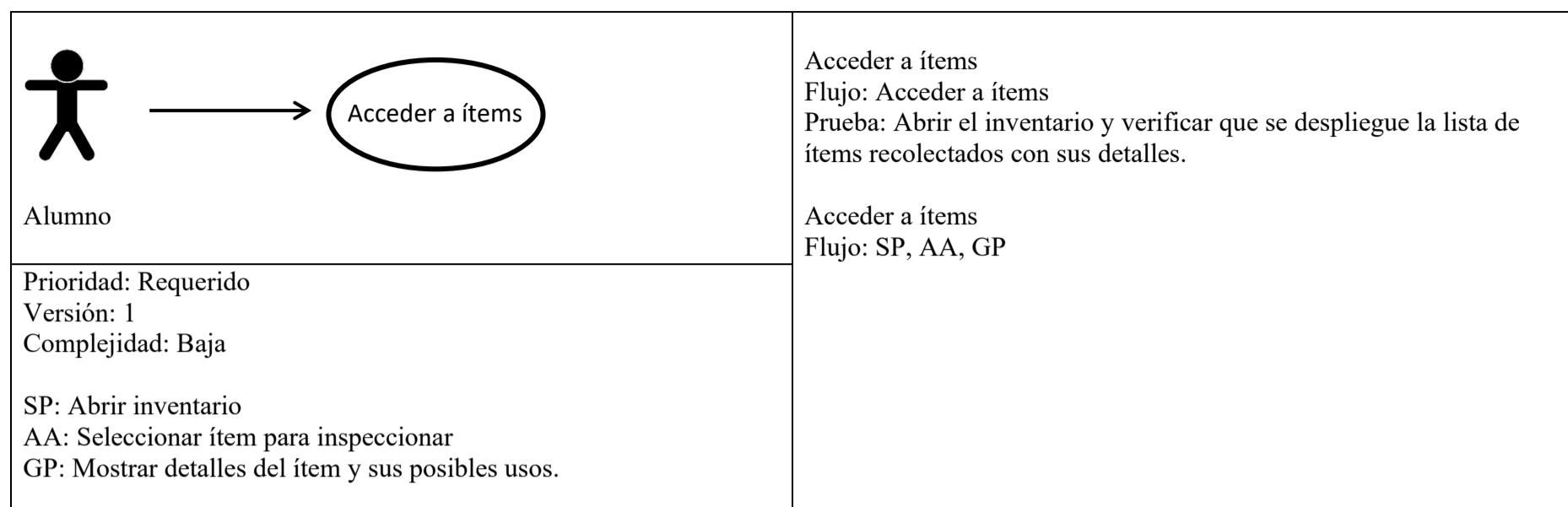
ID	CU-DOC-003	
Nombre	Acceder a estadísticas	
Actores	Docente, Sistema	
Objetivo	Permitir al docente visualizar estadísticas de desempeño de los alumnos.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El docente ha iniciado sesión con permisos válidos.	
Flujo Normal	Docente	Sistema
	Accede al panel de estadísticas	
		Solicita y recupera datos de alumnos
	Selecciona grupo o alumno específico	
		Muestra gráficos, porcentajes y respuestas
		Permite exportar o analizar
Flujo alternativo 1	Filtrá por alumno inexistente	
		Muestra mensaje: "No se encontraron datos"
Flujo alternativo 2	Filtrá por periodo sin actividad	
		Muestra advertencia y sugiere otro filtro
Post-condiciones	El docente puede tomar decisiones pedagógicas basadas en los datos.	
Excepciones	Error de conexión con la base de datos o estadísticas incompletas.	

CASO N°17 Iniciar Sesión (Login)

 → Iniciar sesión (Login)	Iniciar sesión (Login) Flujo: Iniciar sesión (Login) Prueba: Ingresar credenciales válidas y acceder al sistema.
Docente - Alumno - Admin Prioridad: Requerido Versión: 1 Complejidad: Media SP: Ingresar datos de usuario (email y contraseña) AA: Validar credenciales contra la base de datos GP: Crear sesión activa y redirigir al menú principal según el rol	Iniciar sesión (Login) Flujo: SP, AA, GP

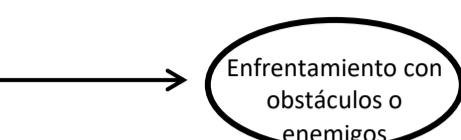
ID	CU-ALUDOADM-001	
Nombre	Iniciar sesión	
Actores	Admin, Docente, Alumno, Sistema	
Objetivo	Permitir a los usuarios iniciar sesión en el sistema para acceder a sus funcionalidades personalizadas.	
Urgencia	5	
Esfuerzo	5	
Pre-condiciones	El usuario debe tener una cuenta previamente registrada en el sistema. El sistema debe estar en línea y operativo.	
Flujo Normal	Alumno	Sistema
	Accede a la pantalla de inicio de sesión	
		Muestra el formulario de login (usuario y contraseña)
	Ingresar usuario y contraseña válidos	
		Verifica las credenciales en la base de datos
		Autentica al usuario y genera token de sesión
		Redirige al usuario al menú principal correspondiente según su rol
Flujo alternativo 1	Muestra mensaje: "Credenciales inválidas. Intente nuevamente"	
Flujo alternativo 2	Muestra mensaje: "Usuario no registrado. ¿Desea crear una cuenta?"	
Flujo alternativo 3	Muestra mensaje: "Error en el servidor. Intente más tarde"	
Post-condiciones	El usuario inicia sesión y accede a sus funcionalidades según su tipo de rol.	
Excepciones	Token inválido Interrupción de la conexión con el backend Ataques por fuerza bruta (protegido mediante bloqueo temporal)	

CASO N°18 Acceder a ítems



ID	CU-ALU-011	
Nombre	Acceder a ítems	
Actores	Alumno, Sistema	
Objetivo	Permitir al alumno ingresar al inventario para visualizar los ítems recolectados, consultar su nombre, tipo, descripción y cantidad disponible, sin generar ningún efecto en el juego.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	El alumno debe haber recolectado al menos un ítem durante la exploración del juego.	
Flujo Normal	Alumno	Sistema
	Abre el menú de inventario	
		Muestra la lista de ítems disponibles
	Navega por la lista de ítems	
	Selecciona un ítem	
		Muestra detalles del ítem (nombre, descripción, tipo, cantidad)
	(No realiza ninguna acción adicional)	
Flujo alternativo 1	El alumno no ha recolectado ítems	
		El sistema muestra el mensaje: "Inventario vacío"
Flujo alternativo 2	El alumno selecciona un ítem inválido o dañado	
		El sistema muestra advertencia: "Ítem no disponible"
Flujo alternativo 3	El sistema tarda en cargar los datos	
		Muestra pantalla de carga o reintento
Post-condiciones	El alumno accede correctamente a la información de los ítems y puede prepararse para usarlos.	
Excepciones	Fallos en la base de datos o error en la visualización del inventario.	

Caso N°19 Enfrentamiento con obstáculos o enemigos

 Alumno	 Enfrentamiento con obstáculos o enemigos	Enfrentamiento con obstáculos o enemigos Flujo: Enfrentamiento con obstáculos o enemigos Prueba: Colisionar con un enemigo u obstáculo en el mapa y verificar que se active el evento correspondiente (daño o bloqueo de paso).
Prioridad: Requerido Versión: 1 Complejidad: Media SP: Detectar colisión con enemigo u obstáculo AA: Activar secuencia de enfrentamiento o penalización GP: Aplicar efecto al personaje y actualizar estado		Enfrentamiento con obstáculos o enemigos Flujo: SP, AA, GP

ID	CU-ALU-012	
Nombre	Enfrentamiento con obstáculos o enemigos	
Actores	Alumno, Sistema	
Objetivo	Permitir que el alumno interactúe con enemigos u obstáculos en el entorno, desencadenando una penalización o secuencia de desafío..	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	El alumno debe estar en una sesión activa del juego y haberse desplazado por el mapa hasta un obstáculo o enemigo.	
Flujo Normal	Alumno	Sistema
	Se desplaza por el mapa	
	se encuentra con un obstáculo o enemigo	
		Detecta la colisión con el objeto o enemigo
		Inicia secuencia de enfrentamiento o aplica penalización (daño, retroceso, etc.)
		Actualiza el estado del personaje en tiempo real
Flujo alternativo 1	Realiza movimiento evasivo a tiempo	
		Cancela daño y mantiene estado del personaje
Flujo alternativo 2	Se acerca a un enemigo visible	
		No detecta interacción y no inicia el enfrentamiento
Post-condiciones	El sistema actualiza correctamente el estado del personaje tras el enfrentamiento o penalización.	
Excepciones	Error en detección de colisión o glitch que permite evitar efectos del obstáculo/enemigo.	

Prioridad de Requerimientos

	Urgencia					
		1-Baja	2-Menor	3-Moderada	4-Alta	5-Obligatoria
Impacto	5-Muy alto				CU-11 CU-12	CU-17
	4-Alto			CU-18 CU-16	CU-19	
	3-Medio		CU-5	CU-15	CU-1	CU-2 CU-3
	2-Bajo		CU-7	CU-10	CU-4 CU-6	CU-9
	1-Muy bajo			CU-8	CU-13 CU-14	

4. Requisitos No Funcionales

Compatibilidad multiplataforma: El videojuego debe ejecutarse de manera fluida en ordenadores de diferentes gamas, priorizando el bajo consumo de recursos para garantizar su uso en contextos escolares con equipos limitados.

Facilidad de mantenimiento y actualización: El sistema debe permitir incorporar nuevos niveles, preguntas, ítems o personajes sin necesidad de reestructurar el código base.

Estética atractiva: El juego debe tener un diseño visual coherente, centrado en el estilo pixel art, que sea accesible y amigable para estudiantes de diversos niveles.

Modularidad del contenido: Las preguntas y los ítems deben poder cargarse de forma dinámica desde una base de datos para permitir personalización por parte de docentes.

Conecividad opcional: El juego debe ser funcional tanto en modalidad en línea como offline (para versiones futuras portables).

Requisitos de Desempeño

Rendimiento en tiempo real: Las acciones del jugador, como movimiento, uso de ítems o interacción con preguntas, deben reflejarse inmediatamente sin demoras perceptibles.

Carga rápida del juego y sus componentes: Los escenarios, sprites, datos del jugador e ítems deben cargarse de forma eficiente para evitar tiempos de espera innecesarios.

Bajo consumo de CPU/GPU: El juego debe optimizar el uso de recursos para evitar ralentizaciones en máquinas escolares o con hardware limitado.

Resistencia a fallos: El juego debe manejar con gracia errores como caídas de conexión o datos incompletos, permitiendo reintentos o notificaciones al usuario.

Requisitos de Seguridad

Autenticación de usuarios: El sistema debe implementar una capa de autenticación segura para permitir el acceso personalizado de alumnos, docentes y administradores.

Protección de datos personales: Las credenciales, estadísticas y resultados de los jugadores deben estar cifrados y protegidos tanto en tránsito como en almacenamiento.

Gestión de sesiones: Las sesiones deben ser válidas por un tiempo definido y cerrarse automáticamente en caso de inactividad prolongada o múltiples accesos concurrentes.

Registros de actividad: Toda interacción clave del jugador (respuestas, avances, ítems recolectados) debe quedar registrada con fines de análisis pedagógico o recuperación de progreso.

Control de acceso por rol: Las funcionalidades disponibles deben variar según el rol (alumno, docente, admin), con restricciones para evitar cambios no autorizados.

Requisitos de Usabilidad

Interfaz intuitiva: El sistema debe contar con menús y controles simples, accesibles para niños y adolescentes, sin necesidad de formación técnica previa.

Diseño accesible: Los elementos gráficos, colores y textos deben cumplir criterios básicos de accesibilidad visual y cognitiva.

Retroalimentación inmediata: Toda acción del usuario (respuesta, movimiento, selección de ítem) debe generar una respuesta visual clara (sonido, animación, texto).

Compatibilidad con dispositivos de entrada variados: Debe poder jugarse con teclado, mouse o pantalla táctil.

Documentación básica: Se deben ofrecer ayudas visuales, tutoriales iniciales o indicaciones contextuales que orienten al jugador en sus primeras interacciones.

Requisitos de Escalabilidad

Carga de múltiples usuarios: El backend debe permitir que varios estudiantes accedan simultáneamente sin afectar el rendimiento (especialmente en torneos escolares o actividades grupales).

Estructura modular: El juego debe permitir añadir nuevos mapas, preguntas, ítems o mecánicas con facilidad para que evolucione a versiones más completas.

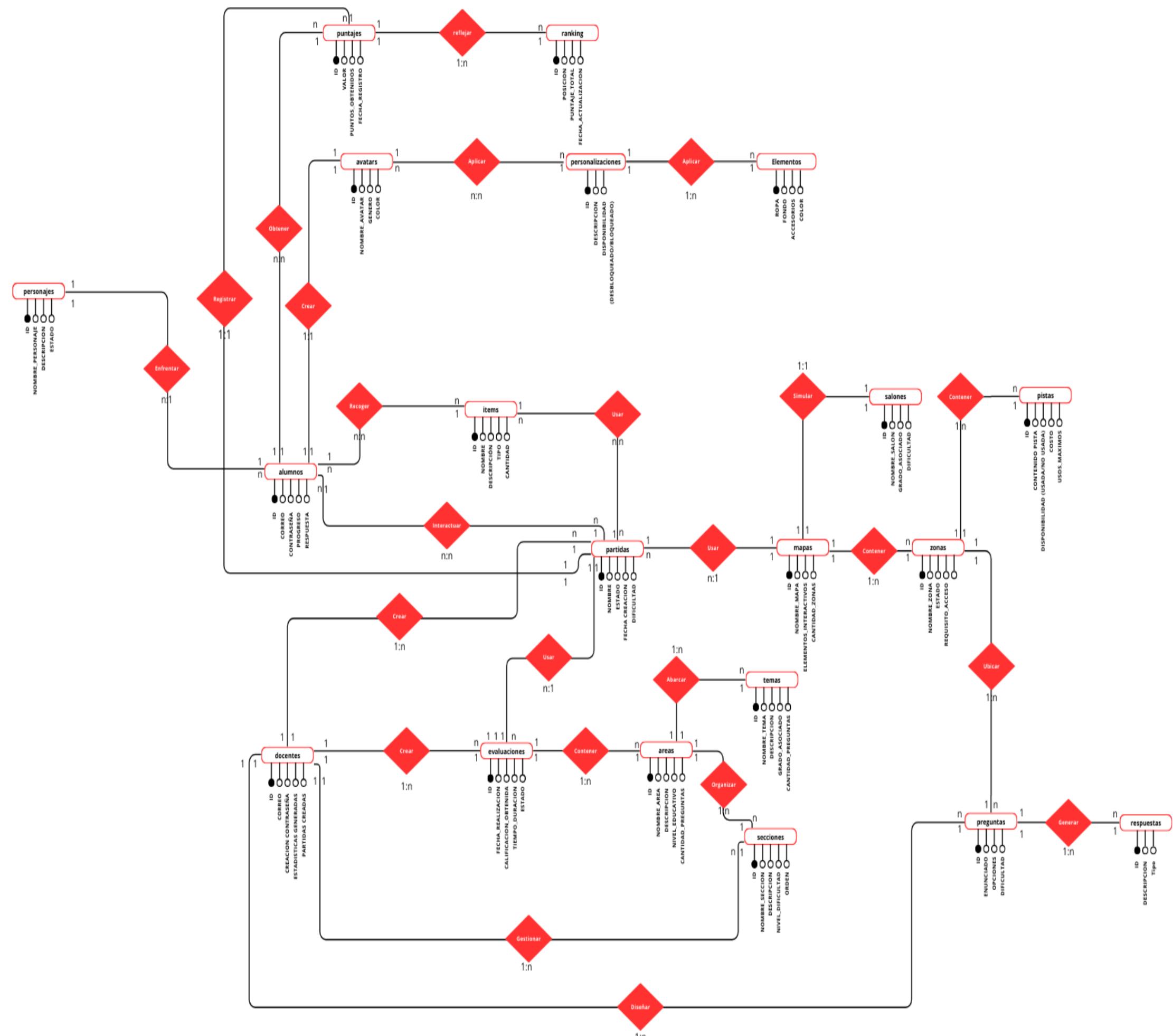
Escalabilidad del servidor: En su versión en línea, el backend debe poder escalar horizontalmente (por ejemplo, usando balanceadores de carga o servicios en la nube).

Soporte para contenido personalizado: En futuras versiones, se debe contemplar la creación de misiones o preguntas por parte de los docentes, escalando no solo en usuarios sino en tipos de contenido.

Integración con servicios externos: Se debe prever la futura conexión con plataformas educativas, analíticas o servicios de IA para generación dinámica de contenido.

5. Modelado E/R

Diagrama de Entidad-Relación



Link: https://www.canva.com/design/DAG0It8jNCw/_d2Z459ijjSrhNBLH23eN-

[Q/edit?utm_content=DAG0It8jNCw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton](#)

Diagrama Relacional

Script de modelo relacional

<https://dbdiagram.io/>

Descripción de Entidades y Relaciones

Entidades:

Usuario

Representa a las personas que acceden al videojuego (alumnos, docentes, administradores).

- **Atributos:** ID (correo electrónico como identificador único), nombre, contraseña, rol, perfil.
- **Relaciones:** Un usuario puede tener varios avatares y participar en múltiples partidas. Los docentes pueden crear evaluaciones y gestionar secciones de estudiantes.

Avatar

Entidad que representa al usuario gráficamente dentro del entorno del videojuego.

- **Atributos:** ID, nombre/avatar, personalización (ropa, colores, accesorios).
- **Relaciones:** Cada avatar pertenece a un usuario y se utiliza en las partidas.

Personaje

Entidad que corresponde a rivales o figuras de la narrativa contra las cuales los alumnos se enfrentan.

- **Atributos:** ID, nombre, tipo, habilidades.
- **Relaciones:** Un personaje puede aparecer en múltiples partidas y enfrentarse a varios usuarios.

Partida

Representa cada sesión de juego en la que participa un estudiante.

- **Atributos:** ID, fecha, estado, duración.
- **Relaciones:** Una partida se desarrolla en un mapa, puede estar asociada a múltiples usuarios y genera puntajes.

Mapa

Espacio general de exploración que estructura la dinámica del juego.

- **Atributos:** ID, nombre, descripción.
- **Relaciones:** Cada mapa se compone de varias zonas y puede ser utilizado en múltiples partidas.

Zona

Subdivisión del mapa que contiene salones, actividades, ítems y pistas.

- **Atributos:** ID, nombre, tipo, estado.
- **Relaciones:** Una zona pertenece a un mapa y puede contener salones, ítems y pistas.

Salón de Aprendizaje

Lugar dentro de una zona donde se desarrollan actividades pedagógicas.

- **Atributos:** ID, nombre, descripción.
- **Relaciones:** Contiene preguntas y está asociado a temas o áreas de conocimiento.

Pregunta

Elemento central de evaluación.

- **Atributos:** ID, enunciado, nivel de dificultad, categoría.
- **Relaciones:** Una pregunta está asociada a un salón de aprendizaje y puede tener múltiples respuestas.

Respuesta

Opciones posibles para cada pregunta, de las cuales solo una es correcta.

- **Atributos:** ID, texto, valor (correcta/incorrecta).
- **Relaciones:** Cada respuesta pertenece a una única pregunta.

Ítem

Recurso estratégico recolectado en las zonas para apoyar el progreso del jugador.

- **Atributos:** ID, nombre, tipo, efecto.
- **Relaciones:** Pertenece a una zona y pueden ser usados por los usuarios en partidas.

Pista

Elemento de ayuda que brinda información adicional al jugador.

- **Atributos:** ID, contenido, tipo (texto, audio, imagen).
- **Relaciones:** Cada pista pertenece a una zona y puede asociarse a preguntas.

Evaluación

Conjunto de preguntas organizadas por un docente.

- **Atributos:** ID, título, fecha de creación.
- **Relaciones:** Pertenece a un área de conocimiento, puede estar asociada a diferentes secciones y asignarse a estudiantes.

Área de Conocimiento

Categoría académica principal.

- **Atributos:** ID, nombre, descripción.
- **Relaciones:** Un área contiene múltiples temas y evaluaciones.

Tema

Subdivisión dentro de un área de conocimiento.

- **Atributos:** ID, nombre, descripción.
- **Relaciones:** Un tema agrupa varias preguntas y actividades.

Puntaje

Registra los resultados de los estudiantes en una partida o evaluación.

- **Atributos:** ID, valor, fecha.
- **Relaciones:** Cada puntaje está vinculado a una partida y a un usuario.

Ranking

Muestra la clasificación de estudiantes según sus puntajes acumulados.

- **Atributos:** ID, posición, referencia a puntajes.
- **Relaciones:** Se alimenta de los puntajes individuales de los usuarios.

Relaciones:

Alumno – Partida:

Un alumno puede interactuar en múltiples partidas, y una partida puede ser interactuada por múltiples alumnos. (N.N)

Alumno – Ítem:

Un alumno puede recoger varios ítems, y un ítem puede ser recogido por varios alumnos. (N.N)

Alumno – Avatar:

Un alumno puede crear un único avatar, y un avatar es creado por un único alumno. (1.1)

Avatar – Personalización:

Un avatar puede aplicar varias personalizaciones, y una personalización puede ser aplicada por varios avatares. (N.N)

Personalización – Elemento:

Una personalización puede aplicar varios elementos, y un elemento puede ser aplicado por una única personalización. (1.N)

Alumno – Puntaje:

Un alumno puede obtener varios puntajes, y un puntaje puede ser obtenido por varios alumnos. (N.N)

Puntaje – Ranking:

Un puntaje se puede reflejar en un ranking, y un ranking puede reflejar múltiples puntajes. (1.N)

Alumno – Personaje:

Un alumno puede enfrentar a varios personajes, y un personaje puede ser enfrentado por un único alumno. (N.1)

Docente – Evaluación:

Un docente puede crear varias evaluaciones, y una evaluación es creada por un único docente. (1.N)

Docente – Partida:

Un docente puede crear varias partidas, y una partida es creada por un único docente. (1.N)

Docente – Pregunta:

Un docente puede diseñar varias preguntas, y una pregunta es diseñada por un único docente. (1.N)

Partida – Puntaje:

Una partida registra un puntaje, y un puntaje es registrado en una partida. (1.1)

Partida – Mapa:

Una partida puede usar un mapa, y un mapa puede ser usado en varias partidas. (N.1)

Partida – Ítem:

Una partida puede usar varios ítems, y un ítem puede ser usado en varias partidas. (N.N)

Mapa – Salón:

Un mapa puede simular un salón, y un salón es simulado por un mapa. (1.1)

Mapa – Zona:

Un mapa puede contener varias zonas, y una zona está contenida en un único mapa. (1.N)

Zona – Pista:

Una zona puede contener varias pistas, y una pista está contenida en una única zona. (1.N)

Zona – Pregunta:

Una zona puede ubicar varias preguntas, y una pregunta está ubicada en una única zona. (1.N)

Pregunta – Respuesta:

Una pregunta puede generar varias respuestas, y una respuesta es generada por una única pregunta. (1.N)

Reglas de Integridad Referencial

Colecciones (NoSLQ)

6. Anexos

Diagramas Adicionales

Referencias

Etapa 2: Persistencia de Datos con Backend

7. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

8. Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

9. Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

10. Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

11. Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

12. Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapa 3: Consumo de Datos y Desarrollo Frontend

13. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

14. Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

15. Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

16. Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

17. Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

18. Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

19. Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

20. Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend

ANEXOS

Diagramas UML